

REQUEST FOR ACCESS OF ABANDONED APPLICATION UNDER 37 CFR 1.14(a)

In re Application of

Baerlein

Application Number

Filed

07/719507

Mar 20, 1990

Group Art Unit

Examiner

RECEIVED

MAR 29 2001

File Information Unit

Paper No. *#47*

Assistant Commissioner for Patents
Washington, DC 20231

I hereby request access under 37 CFR 1.14(a)(3)(iv) to the application file record of the above-identified ABANDONED application, which is: (CHECK ONE)

- (A) referred to in United States Patent Number 5694603 column _____
- (B) referred to in an application that is open to public inspection as set forth in 37 CFR 1.11, i.e., Application No. _____ filed _____ on page _____ of paper number _____
- (C) an application that claims the benefit of the filing date of an application that is open to public inspection, i.e., Application No. _____ filed _____ or
- (D) an application in which the applicant has filed an authorization to make open the complete application to the public.

Please direct any correspondence concerning this request to the following address:

Vivianne Pondexter

Signature

Vivianne Pondexter

Typed or printed name

3/29/01

Date

FOR PTO USE ONLY

Approved by: *gln*
(Initials)

Unit *P-A-4*



US005694603A

United States Patent [19]

Reiffin

[11] Patent Number: 5,694,603
[45] Date of Patent: Dec. 2, 1997

[54] COMPUTER MEMORY PRODUCT WITH PREEMPTIVE MULTITHREADING SOFTWARE

[76] Inventor: Martin G. Reiffin, 5439 Blackhawk Dr., Danville, Calif. 94526

[21] Appl. No.: 496,282

[22] Filed: Mar. 20, 1990

Related U.S. Application Data

[63] Continuation of Ser. No. 425,612, Sep. 28, 1982, abandoned, and Ser. No. 719,507, Apr. 3, 1985, abandoned.

[51] Int. Cl. 6 G06F 9/46

[52] U.S. Cl. 395/677

[58] Field of Search 395/800, 375, 395/650, 677, 678

[56] References Cited

U.S. PATENT DOCUMENTS

3,614,745	10/1971	Poduin	395/650
4,084,228	4/1978	Dufond et al.	395/650
4,383,307	5/1983	Gibson, III	395/575
4,513,391	4/1985	Maddock	395/146
4,671,684	6/1987	Kojima et al.	400/63
4,724,285	2/1988	Lefler et al.	178/21
4,773,009	9/1988	Kucera et al.	364/419
4,887,212	12/1989	Zamora et al.	364/419

OTHER PUBLICATIONS

Hiromoto, Robert, *Parallel-processing a large scientific problem*, AFIPS Press, 1981, pp. 235-237.
Ousterhout, John K., *Scheduling techniques for Concurrent Systems*, IEEE, 1982, pp. 22-30.

Andrews, Gregory R., *Synchronizing Resources*, ACM Transactions on Programming Languages and Systems, vol. 3, No. 4, Oct. 1981, pp. 405-430.

Collin, A. J. T., *The Implementation of STAB-I*, Software—Practice and Experience, vol. 2, 1972, pp. 137-142.

Artym, Richard, *The STAB Multiprocessing Environment for CYBA-M*, Software—Practice and Experience, vol. 12, 1982, pp. 323-329.

Treleaven et al., *Combining Data Flow and Control Flow Computing*, The Computer Journal, vol. 25, No. 2, 1982, pp. 207-217.

Duffie, C. A. III, *Task Scheduling Algorithm for a Teleprocessing Communications Controller*, IBM Technical Disclosure Bulletin, vol. 16, No. 10, Marcy 1974, pp. 3349-3352.

Hoare, C. A. R., *Towards a Theory of Parallel Programming*, Operating Systems Techniques, Proceedings of a Seminar held at Queen's University, Belfast, 1972, Academic Press, 1972, pp. 61-71.

Cheriton, David Ross, *Multi-Process Structuring and the Thoath Operating System*, Doctoral Thesis, University of Waterloo, 1978.

Redell et al., *Pilot: An Operating System for a Personal Computer*, Communications of the ACM, Feb. 1980, vol. 23, No. 2, pp. 81-92.

Lampson et al., *Experience with Processes and Monitors in Mesa*, Communications of the ACM, Feb. 1980, vol. 23, No. 2, pp. 105-117.

Hughes, Lawrence, E., "System Programming Under CP/M-80," 1983, pp. 109-112 and 127-138.

Primary Examiner—Richard L. Ellis

[57] ABSTRACT

A multithreading computer system for the preemptive asynchronous concurrent execution of a plurality of instruction threads of a multithreaded application program. As an illustrative example of one of the many uses of the invention, the disclosed application program comprises a compiler thread and an editor thread. The compiler processes the source code while the programmer edits the source code at the keyboard. An interrupt service routine repeatedly activates the editor thread to provide a preemptive asynchronous editing process for either entry of new source code or modification of the previously entered code, while the compiler thread concurrently processes the source code during the time intervals between keystrokes. The interrupt service routine may be activated either by a keyboard interrupt or periodically by a clock at predetermined time intervals.

41 Claims, 9 Drawing Sheets

